

# Android Chat Application

## By Fahad Alkamli

### S3498306

## Introduction

The purpose of this project is to implement a functioning chat app based on a cloud architecture where users can create account, add friends and start a chat with a friend. The motivation behind my project is to get to know how apps we use on a daily basis work and what technologies are being used in the back-end. Not all of us will get the chance to work in a big project like implementing an app similar to WhatsApp or Line. Therefore, we miss the chance on learning how real life applications work and the number of hours is being put to make such apps successful.

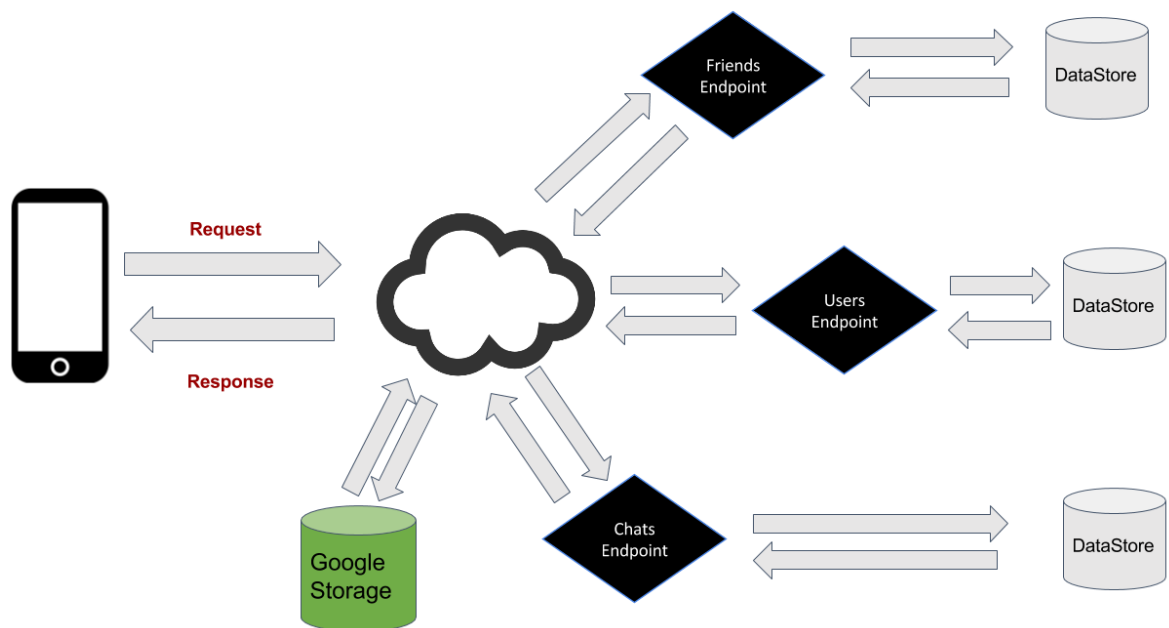
# Contents

1. Introduction .....	1
2. Contents .....	2
3. Links .....	3
4. Software Design/Architecture .....	3
5. Endpoints	
5.1. Users Endpoint .....	4
5.2. Friends Endpoint .....	4
5.3. Chats Endpoint .....	5
6. Google Storage .....	5
7. Implementation - Developer Manual	
7.1. Importing Endpoints Project .....	6-13
7.2. Importing Android Chat App .....	14
8. User Manual .....	7
8.1. User Interface .....	
8.1.1. Welcome Page .....	
8.1.2. Registration .....	
8.1.3. Login .....	
8.1.4. Home/Friends List .....	
8.1.5. Add Friend by Email .....	
8.1.6. Add Friend from Contacts .....	
8.1.7. User Settings .....	
8.2. Admin Extra Feature .....	
8.2.1. Access Admin Panel .....	

## Links

- API Explorer:  
[https://apis-explorer.appspot.com/apis-explorer/?base=https://chat-api-s3498306.appspot.com/\\_ah/api#p/](https://apis-explorer.appspot.com/apis-explorer/?base=https://chat-api-s3498306.appspot.com/_ah/api#p/)
- 

## Software Design/Architecture



## Users Endpoint:

This endpoint will deal with requests related to the user and will give a response back in JSON format.

---

`userendpoint.averageUnreadMessages`

---

---

`userendpoint.getTotalChatHistory`

---

---

`userendpoint.getTotalUsers`

---

---

`userendpoint.getUser`

---

---

`userendpoint.insertUser`

---

---

`userendpoint.listUser`

---

---

`userendpoint.makeAdmin`

---

---

`userendpoint.refreshToken`

---

---

`userendpoint.removeUser`

---

---

`userendpoint.updateUser`

---

## Friends Endpoint:

This endpoint will take requests related to friendship operations and will respond with a JSON format.

---

`friendendpoint.friendsExists`

---

---

`friendendpoint.getFriends`

---

---

`friendendpoint.insertFriend`

---

---

`friendendpoint.listFriend`

---

---

`friendendpoint.removeFriend`

---

## Chats Endpoint:

This endpoint will take requests related to chat operations and will respond with a JSON format.

---

`chatendpoint.getAllChats`

---

`chatendpoint.insertChat`

---

`chatendpoint.removeChat`

---

`chatendpoint.testMessages`

---

`chatendpoint.topTenMostCommonWords`




---

## Google Storage:

I used Google storage to store users profile pictures. Each user can upload a picture from their device to the cloud. This process will produce a public URL to the picture, the URL will be stored in the user's profile information in the user's data-store.

[Buckets](#) / chat-api-s3498306.appspot.com

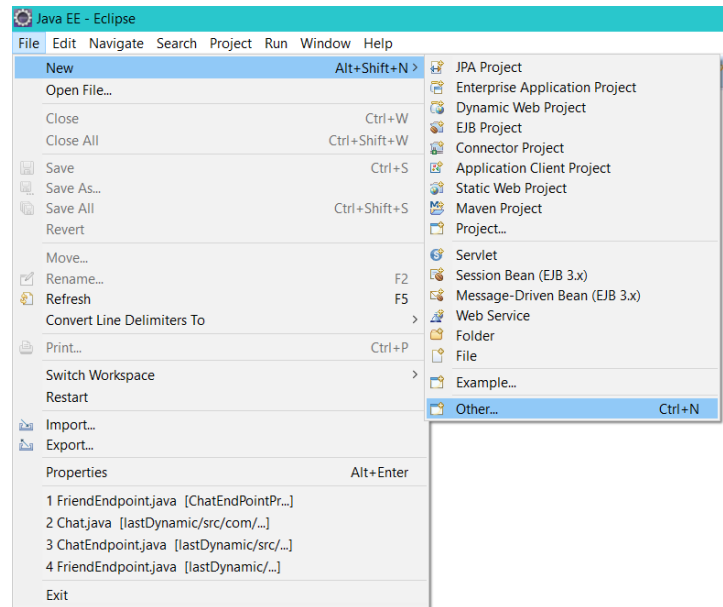
 Filter by prefix...

<input type="checkbox"/>	Name	Size	Type	Last modified	Share publicly
<input type="checkbox"/>	 profilePicture-alkamli@hotmail.com.jpg	99.99 KB	—	16/09/2016, 02:47	<a href="#">Public link</a>
<input type="checkbox"/>	 profilePicture-d0l1@hotmail.com.jpg	80.53 KB	—	27/09/2016, 19:16	<a href="#">Public link</a>
<input type="checkbox"/>	 profilePicture-jack@hotmail.com.jpg	17.47 KB	—	10/09/2016, 17:19	<a href="#">Public link</a>

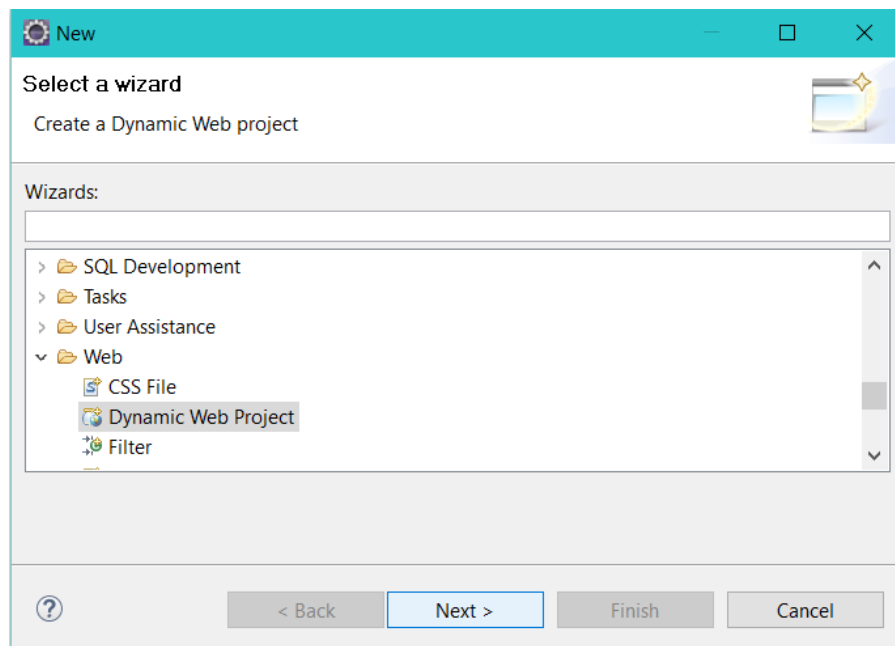
# Implementation - Developer Manual

- **Importing Endpoints Project**

1- Open Eclipse and click on “File” from the menu select “New ” then “Other”



2- Under “Web” Select “Dynamic Web project”



- 3- Name the project “ChatEndPointProject” and set the configuration to “Google App Engine with JPA”

**New Dynamic Web Project**

Dynamic Web Project  
Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: ChatEndPointProject

Project location  
☒ Use default location  
Location: C:\Users\d011\OneDrive\ClassesBackup\workspace\ChatEndPointProject [Browse...](#)

Target runtime  
Google App Engine [New Runtime...](#)

Dynamic web module version  
2.5

Configuration  
Google App Engine with JPA [Modify...](#)  
Adds JPA facet into Google App Engine default configuration.

EAR membership  
☐ Add project to an EAR  
EAR project name: EAR [New Project...](#)

Working sets  
☐ Add project to working sets  
Working sets: [Select...](#)

[?](#) < Back Next > **Finish** Cancel

- 4- Click next and next next next . Here you will see the package name change it to be “com.fahad.chatEngine” and click finish.

**New Dynamic Web Project**

Google App Engine  
⚠ You won't be able to deploy to Google without valid Project ID.

Project  
Package: com.fahad.chatEngine

Deployment  
Project ID: [My projects...](#)  
Version: 1 [Existing versions...](#)  
Module:

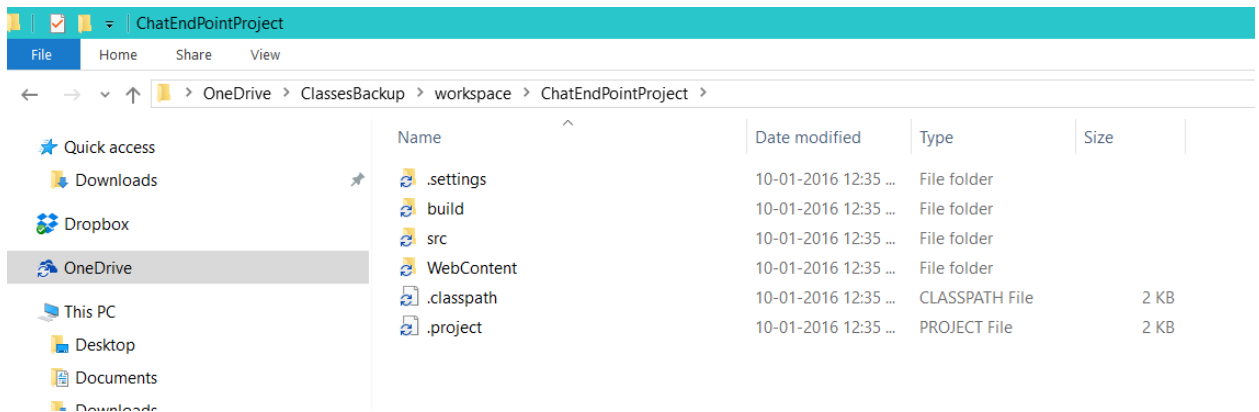
Deployment Options  
☒ Enable jar splitting  
☐ Package WEB-INF/classes as a jar  
☐ Retain staging directory

Sample Code  
☒ Generate project sample code

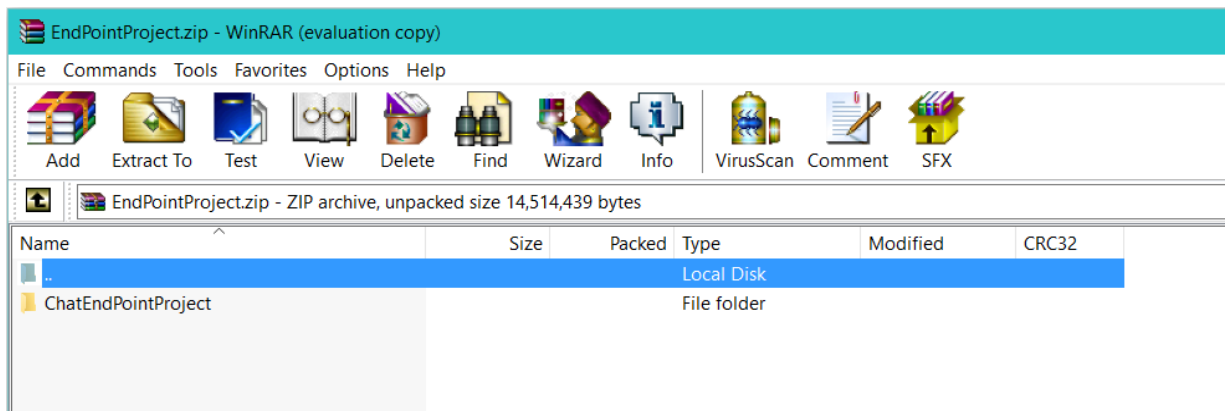
Google APIs  
☐ Open Google API Import Wizard upon project creation

**Finish**

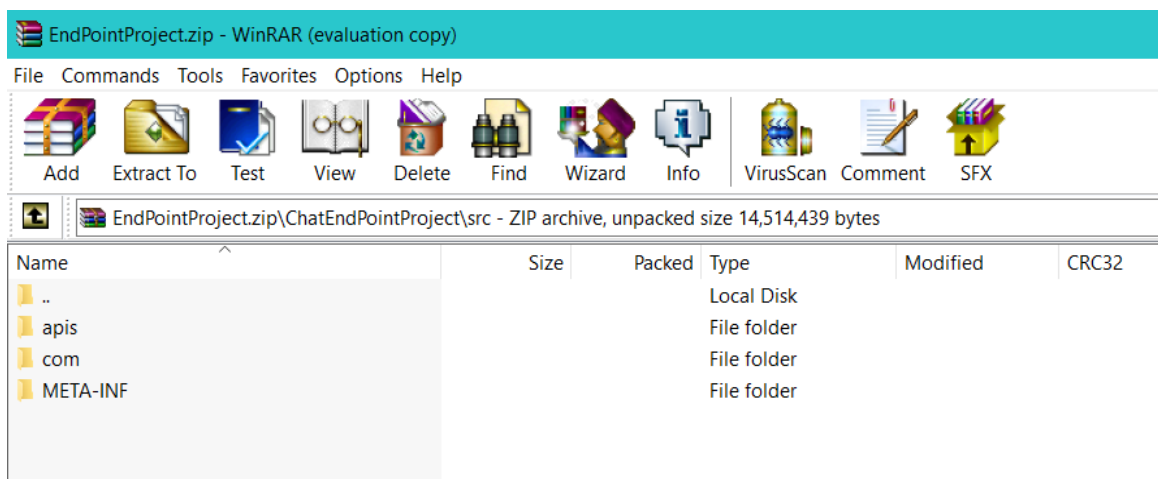
5- Locate the project in your workspace folder and go next to the folder “src”



6- Now is time to open the zip file called “ChatEndPointProject.zip” using any zipping program , I am using RAR Here.

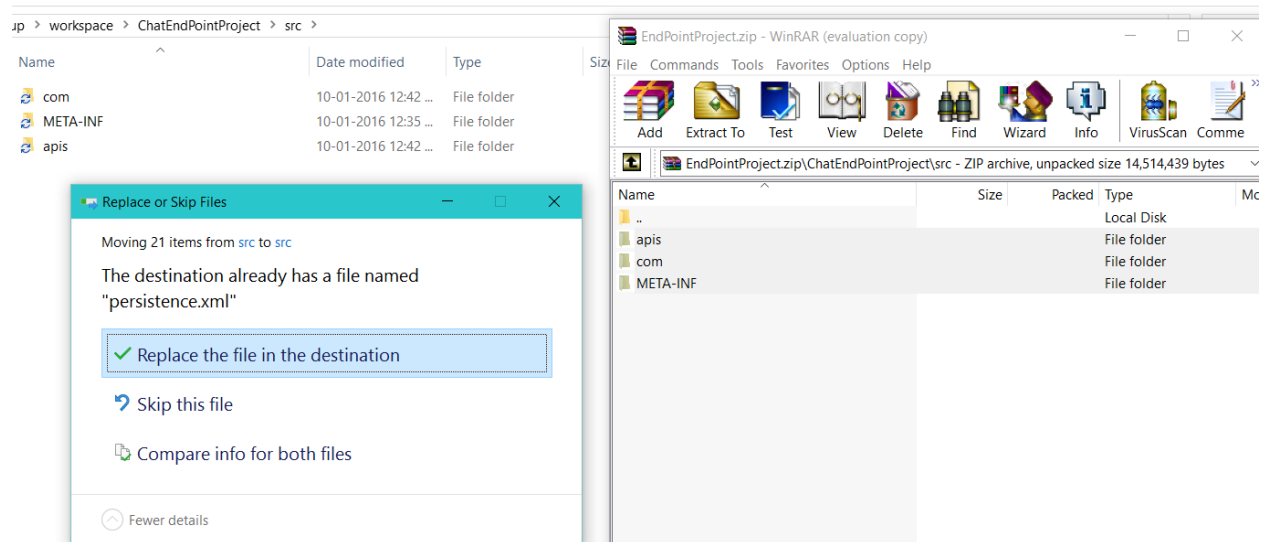


7- Now go inside the folder called “ChatEndPointProject” and then go also inside the folder “src” , now extract the following folders into your project ,this will replace your project’s folders with the zip file’s folders.

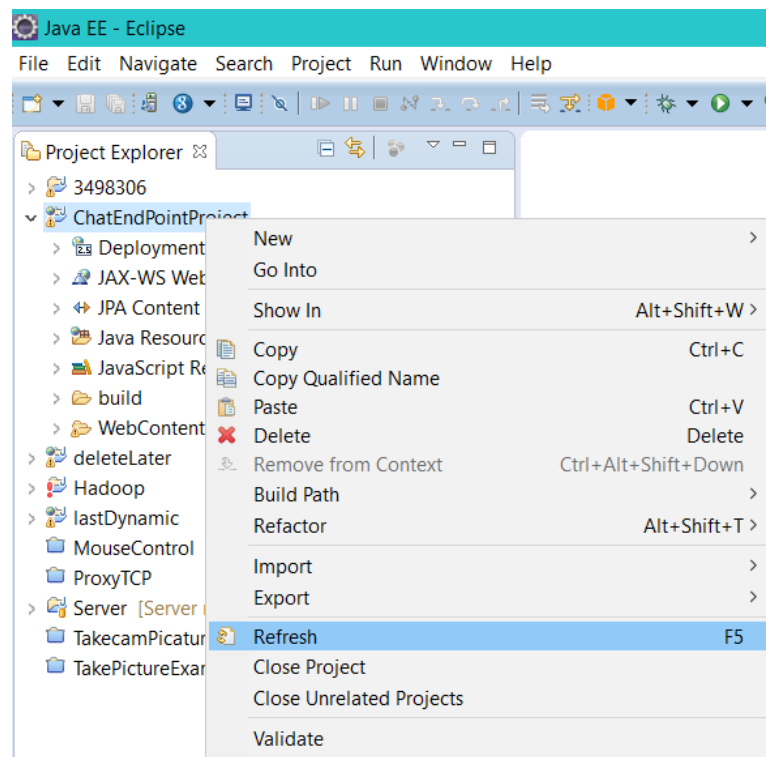




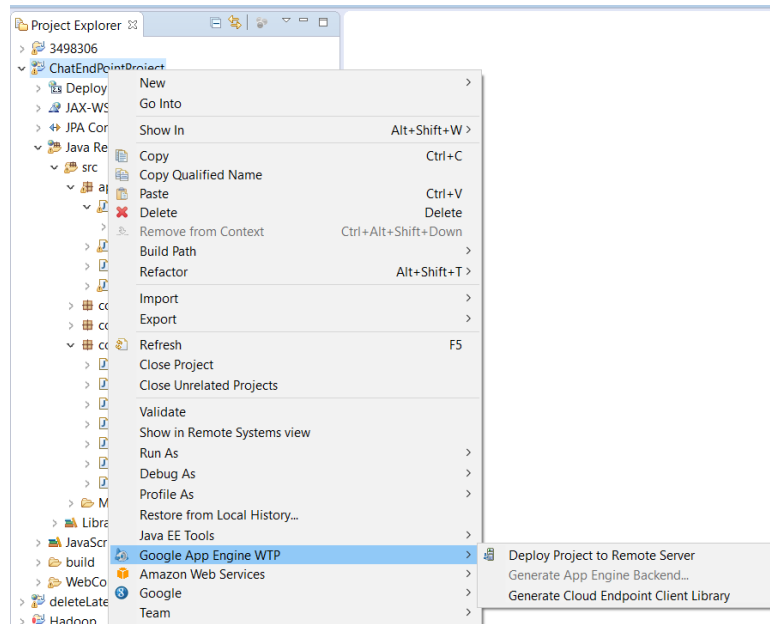
8- Now you will be prompted to confirm the replacement select “Replace the file in the destination”



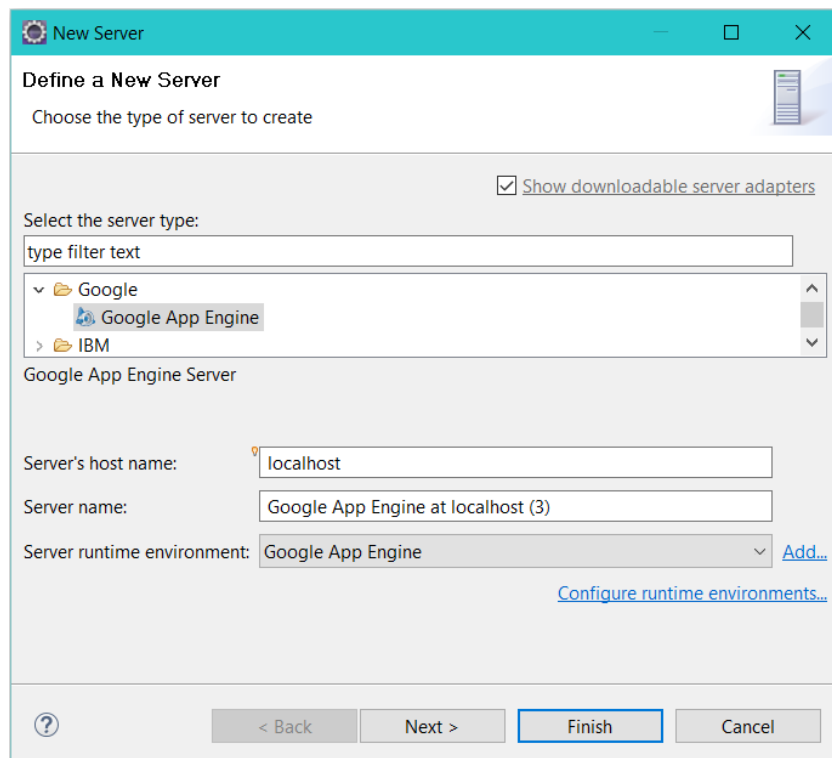
9- Go back to Eclipse and select your project and right click , then click “Refresh” .



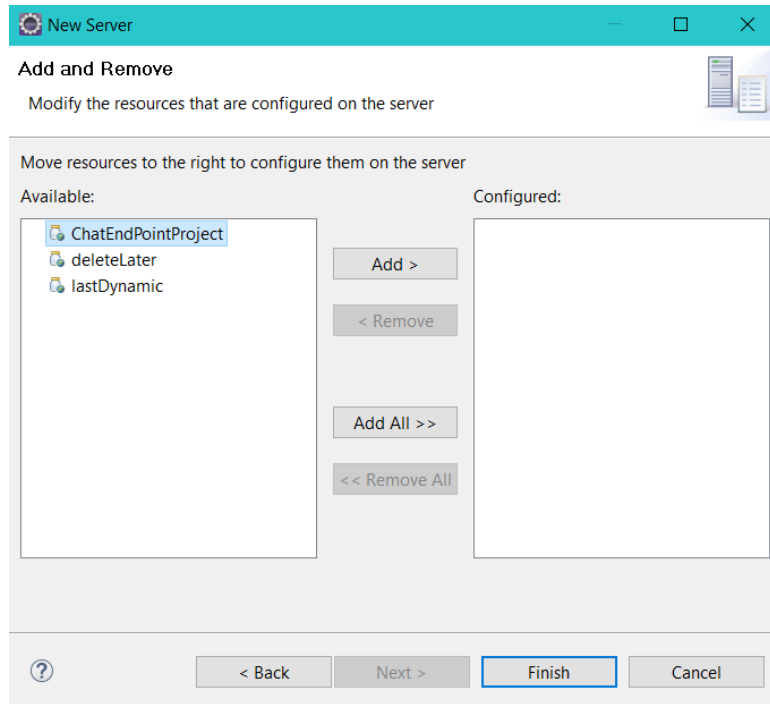
10-To Deploy the project to your Google cloud, right click on your project and locate “Google App Engine WTP” and then click on “Deploy Project to Remote Server”



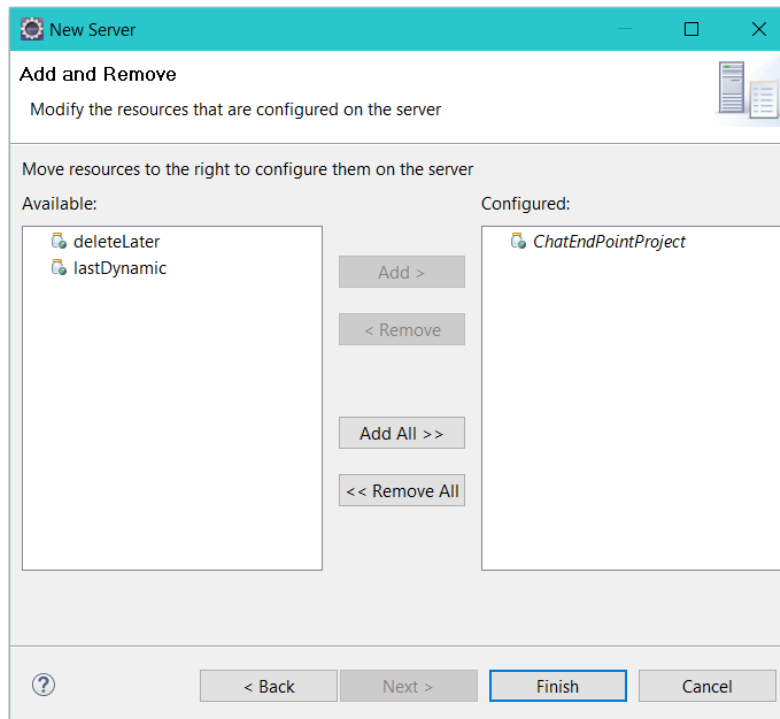
11-from the menu make sure that the selected Server type is “Google App Engine” and click next and follow it with another next.



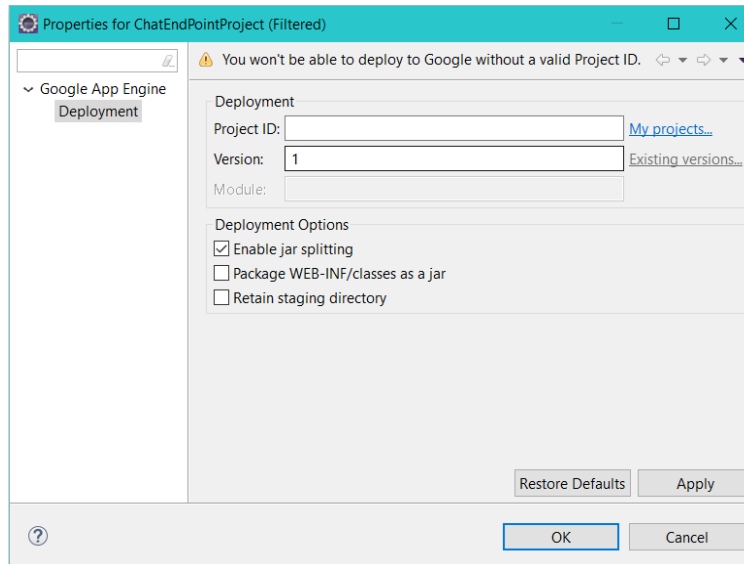
12-Now select your project from the left side and click on “Add” , the project will move from the left-side to the right-side



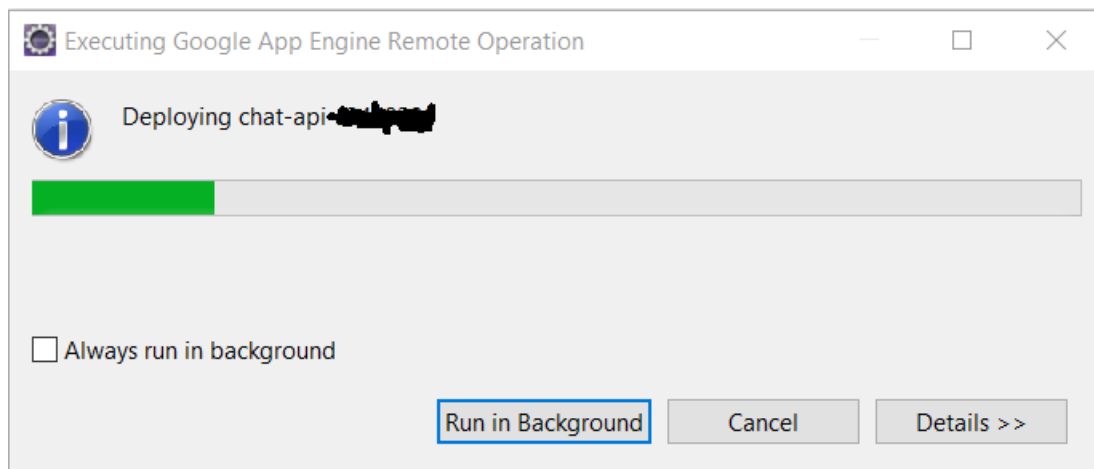
13-Click finish



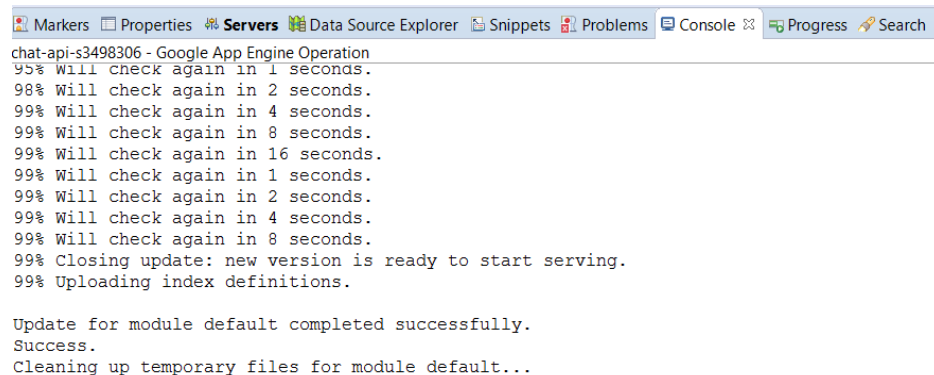
14-Next you will be promoted to enter your Google cloud project ID, after that click apply and ok



15-Next you will see a window showing the deployment

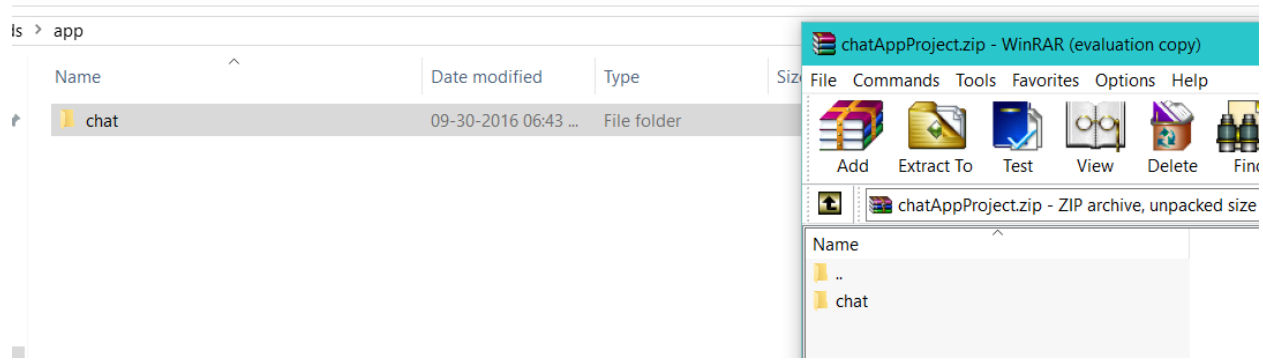


16-If you would like to see the progress of the deployment, then go to the console and you will see if the deployment was successful or not.

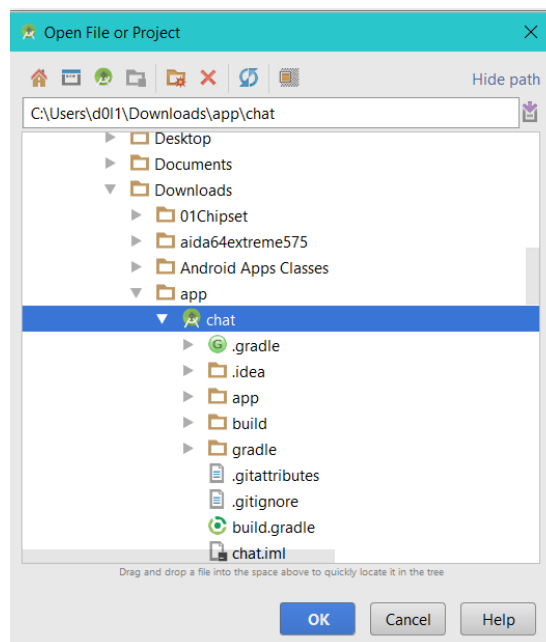


## • Importing Android Chat App

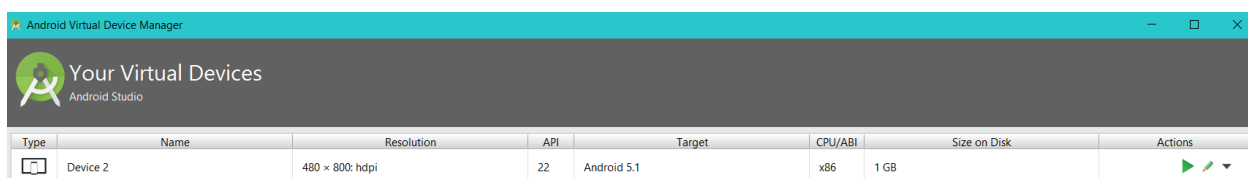
1- Extract the zip file called “chatAppProject.zip”



2- Open Android Studio and click on “File” then “Open”, next locate your folder and click on the Green icon next to “Chat” and then Ok



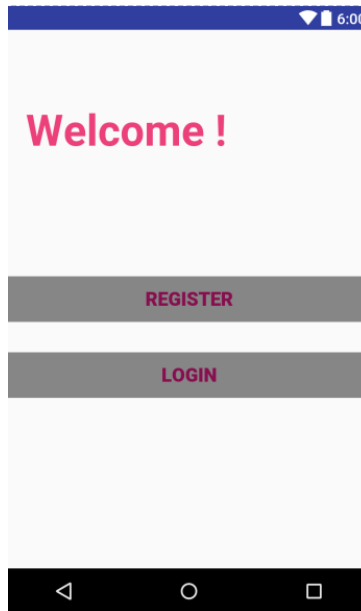
3- Android Studio will open the project and you can run it on any emulator with a minimum API of 22.



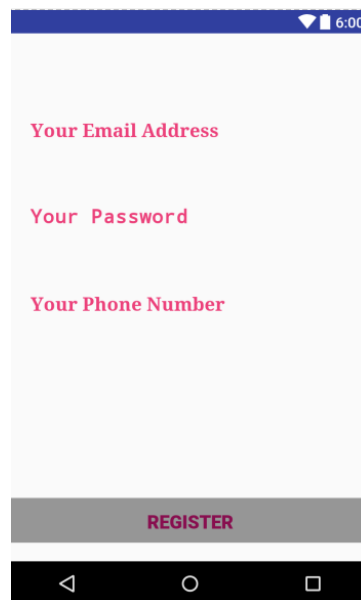
# User Manual

## User Interface

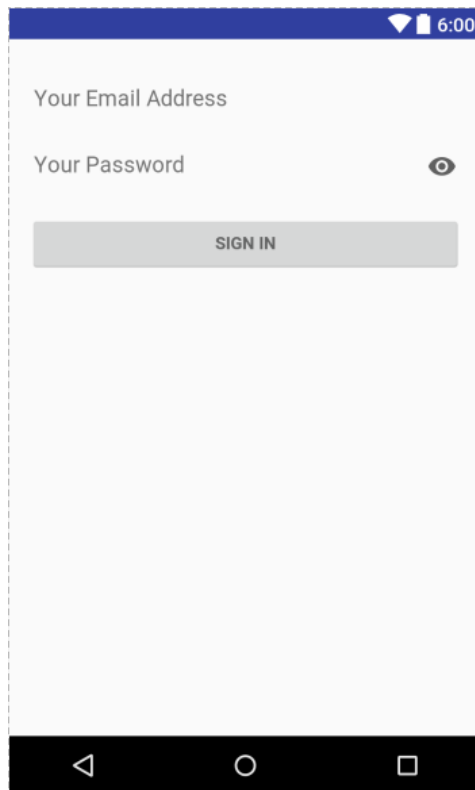
- Welcome Page



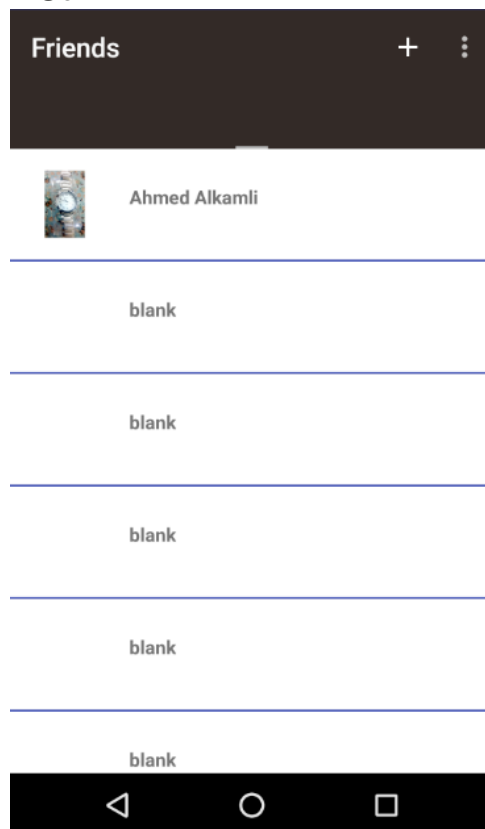
- Registration



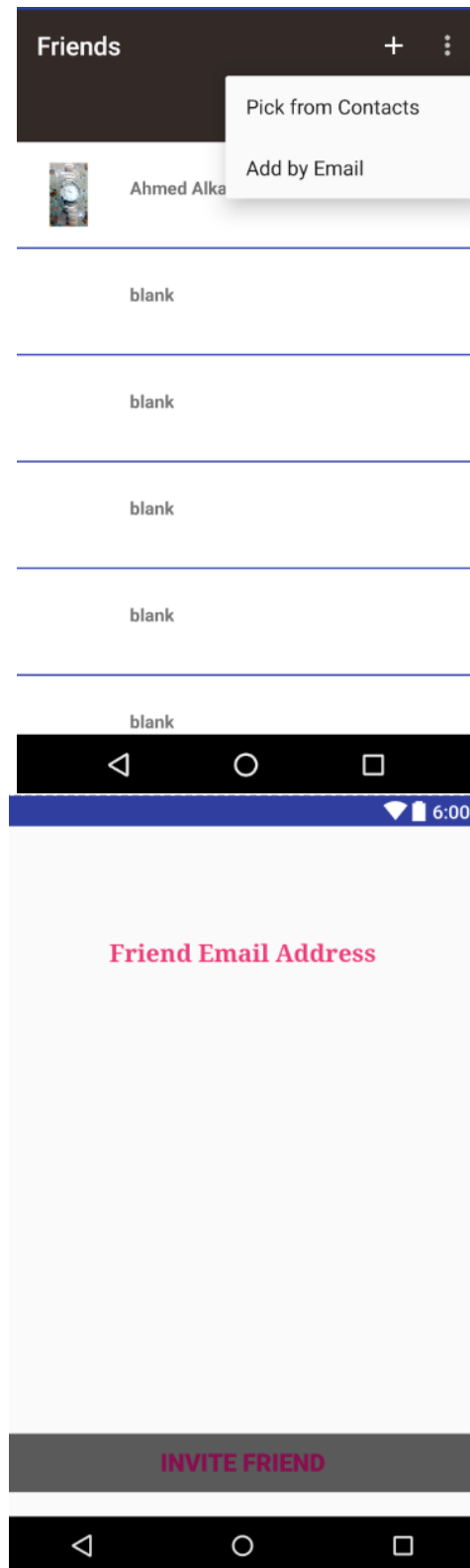
- Login



- Home/Friends List

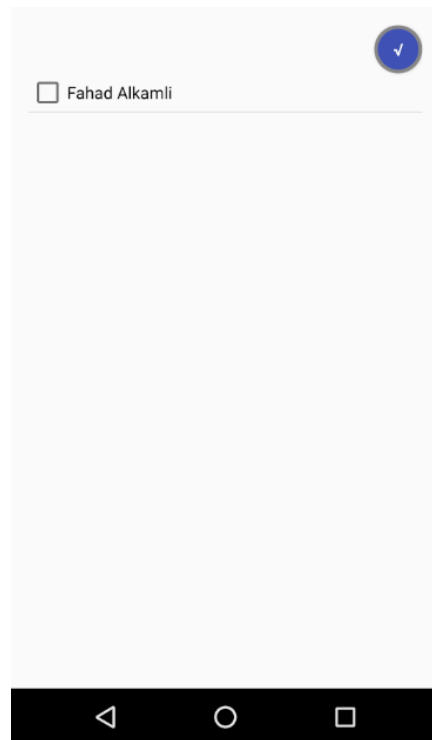


- Add Friend by Email





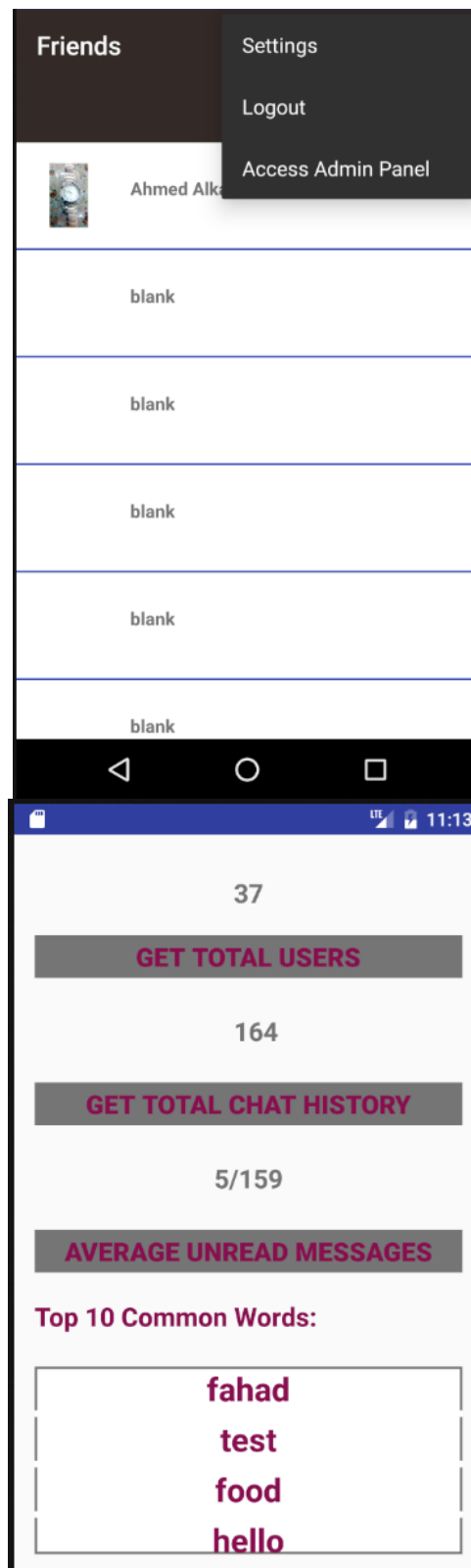
- Add Friend from Contacts



- User Settings



## Admin Extra Feature



## Reference List:

- 1- <http://www.freeiconspng.com/uploads/cloud-icon-29.png>
- 2- <https://d30y9cdsu7xlg0.cloudfront.net/png/74599-200.png>
- 3- okhttp Android Library
- 4- Android Google sdk